Practical assignment 2 (Introduction on neural learning on page 65 of the Grokking Deep Learning textbook)

Consider the following code snippet which implements a very simple form of neural learning:

```
 1: weight = 0.5
 2: goal_pred = 0.8
 3: input = 0.5
 4:
 5: for iteration in range(20):
 6:     pred = input * weight
 7:     error = (pred - goal_pred) ** 2
 8:     direction_and_amount = (pred - goal_pred) * input
 9:     weight = weight - direction_and_amount
10:     print("Error:" + str(error) + " Prediction:" + str(pred))
```

Please modify the code mentioned above in the following manner:

1) Add a variable `minimum_error` variable in line 4 with a very small value, e.g. 1e-6.
2) Adjust the code so that the program would terminate when the error calculated in line 7 is equal to or below the `minimum_error` value in line 4. This means that you must not run the program for a specified number of iterations, but the program must run until the `minimum_error` is found.
3) Add descriptive comments to clarify your code.

At first glance it seems like the perfect solution. You can set the minimum error to a very small amount and the neural network would be able to find the corresponding weight value. Please write a report of no more than 1000 words on why this is not such a good idea in practice. Cite reputable sources to motivate your arguments.